Examen – Complexité (SINA09B)

Durée : 2 heures. *(Barème indicatif)*
Documents : non autorisés.
Important : on vous demande de présenter <u>proprement</u> et <u>clairement</u> vos réponses.
Indications : $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$, et si $\mathbf{A} \leq_m^P \mathbf{B}$ (réduction many-one polynomiale) alors $\mathbf{A} \leq_T^P \mathbf{B}$.

Exercice 1. *Complexité algorithme itératif (4 points)*

Voici un algorithme `algo1` prenant en entrée un entier naturel $\mathtt{n} \in \mathbb{N}$.

```
algo1(n):
    r = 0
    si n est multiple de 3 alors:
        pour i de 1 à n faire:
            pour j de 1 à i faire:
                pour k de 1 à j faire:
                    r = r + 3
    sinon:
        pour i de 1 à n*n faire:
            r = r + 1
    retourner r
```

1. Donner une majoration asymptotique aussi serrée que possible de la complexité dans le pire cas de l'algorithme `algo1` ci-dessus (notation grand-O), en justifiant brièvement votre analyse.
2. Cette complexité est-elle polynomiale ou exponentielle ? Justifier.

Exercice 2. *Complexité algorithme récursif (4 points)*

Voici un algorithme `algo2` prenant en entrée un tableau d'entiers `T` de taille `n`, où `T.length()` retourne la taille du tableau, et `T.pop()` retire le dernier élément du tableau, chacune en temps constant.

```
algo2(T):
    n = T.length()
    si n == 0 alors retourner -1
    si T[n-1] == 42 alors retourner n-1
    T.pop()
    retourner algo2(T)
```

1. Que retourne l'appel `algo2([50,22,8,110,73])` ? Il s'agit d'un pire cas.
2. Soit $C(\mathtt{n})$ la complexité dans le pire cas de `algo2` pour un tableau de taille `n`. Donner une équation de récurrence pour $C(\mathtt{n})$, et sa condition initiale pour $\mathtt{n} = 0$.
3. Donner une solution à votre équation de récurrence, qui vérifie également sa condition initiale pour $\mathtt{n} = 0$. Justifier votre solution en montrant qu'à partir de votre équation de récurrence, en remplaçant $C(\mathtt{n}-1)$ par votre solution pour $\mathtt{n}-1$, on obtient votre solution pour `n`.

Exercice 3. *Classe* NP *(8 points)*

Soient **A** et **B** deux problèmes de décision.

1. Expliquer précisément pourquoi, si **A** est NP-complet et **A** $\leq_T^P$ **B**, alors **B** est NP-difficile.
2. Expliquer précisément pourquoi, si **B** est NP-complet et **A** $\leq_T^P$ **B**, alors **A** $\in$ NP.
3. Expliquer précisément pourquoi, si on démontre que NP $\neq$ coNP, alors on gagne \$ 1 000 000 car on en déduit une réponse à la grande question de savoir si P est égale à NP.

Soient les deux problèmes de décision suivants.

Clique
Entrée : Un graphe non orienté $G = (V, E)$ à n sommets, et un entier $k \in \mathbb{N}$.
Question : G possède-t-il une clique de taille supérieure ou égale à k ?

1/2-Clique
Entrée : Un graphe non orienté $G = (V, E)$ à n sommets.
Question : G possède-t-il une clique de taille supérieure ou égale à $\frac{n}{2}$?

1/4-Clique
Entrée : Un graphe non orienté $G = (V, E)$ à n sommets.
Question : G possède-t-il une clique de taille supérieure ou égale à $\frac{n}{4}$?

4. Démontrer que **1/4-Clique** appartient à la classe NP, en décrivant le certificat, l'algorithmique du vérificateur (déterministe), et le temps de calcul de ce dernier.
5. Démontrer que **1/2-Clique** $\leq_T^P$ **1/4-Clique**.
6. Démontrer que **Clique** $\leq_T^P$ **1/2-Clique**.
   Indication : on peut faire trois constructions différentes, selon si $k < \frac{n}{2}$ ou $k = \frac{n}{2}$ ou $k > \frac{n}{2}$.
7. Sachant que **Clique** est NP-complet, que peut-on déduire des questions précédentes de cet exercice, sur la complexité de **1/2-Clique** et de **1/4-Clique** ? Précisez vos raisonnements.

Exercice 4. *Modélisation (4 points)*

Hamiltonien
Entrée : Un graphe non orienté $G = (V, E)$ à n sommets.
Question : G possède-t-il un cycle hamiltonien ?

Un *cycle hamiltonien* est un cycle de longueur n parcourant une et une seule fois chaque sommet de G.

Nous allons modéliser ce problème à l'aide d'une formule propositionelle (réduction vers **SAT**), en utilisant n^2 variables notées $x_{i,j}$ pour tout $1 \leq i \leq n$ et tout $1 \leq j \leq n$, avec la sémantique que $x_{i,j} = 1$ lorsque le sommet i est en position j dans le cycle Hamiltonien.

1. Quelles clauses faut-il ajouter afin que chaque position contienne un, et un seul, sommet ?
2. Quelles clauses faut-il ajouter afin que chaque sommet apparaisse en une, et une seule, position ?
3. Quelles clauses faut-il ajouter afin que la séquence de sommets soit bien un cycle du graphe G ?